Automated Redistribution Functional Specification

Version 1.1

February 4, 2003

Background and Overview

The management of Provincial elections is based on the administrative division of BC into a number of Electoral Districts (EDs), which are in turn subdivided into Voter Areas (VAs). Periodically, the numbers of EDs and VAs are increased and/or their boundaries are redefined to reflect evolving changes in the geographic distribution of the population.

The redistribution of EDs is done after every second general election, and involves an extensive consultative process, with the EDs being defined on the basis of political and social considerations as well as geographic factors. Because of this, the ED redistribution problem is not really a suitable candidate for the application of automated methods.

The redistribution of VAs is also done periodically, typically several times during the period between ED redistributions. However, the considerations for defining the VA numbers and boundaries are much more objective than for EDs. The major criteria for defining VAs are that they:

- should have voter populations between specified lower and upper limits;
- should be reasonably compact;
- should not be divided by major physical barriers that would inhibit access of all the voters in the VA to the voting place designated for the VA, and
- should not span major administrative boundaries (e.g., municipal boundaries).

The nature of these criteria is highly favorable for the application of automated methods for defining the VA boundaries.

In a previous stage of the INDEA project, BCS examined the general redistribution problem, surveyed the literature and the state of software, and designed and implemented a prototype algorithm for redistribution, written in IDL[1]. The approach taken was to use a selected set of INDEA layers to generate a coverage of polygonal atoms for the ED, and then to build up the VAs using these atoms as building blocks. At each stage, an initial atom was chosen as the "seed" for the new VA, and adjacent atoms were then combined

[1] Interactive Data Language, produced by Research Systems, Inc. (now ITT Visual Information Solutions).

with the growing VA until the population criteria were satisfied. This prototype produced very promising results when applied to real data, especially for urban areas.

In the present project, the INDEA Data Manager (IDM) will be extended to provide an environment for automated redistribution of VAs, using the approaches developed in the IDL prototype.

The IDM redistribution environment will consist of three main components:

- An interface and application for generating an atom coverage for a specified ED and Boundary set. This application forms the atoms using standard and optional layers from INDEA, and results in the production of a mapEdit[2] file and its transfer to a local directory on the client machine.
- An interface and application for automatically generating a number of candidate VA tilings for the ED. This application run within the IDM and uses a C/C++ version of the above IDL algorithm to group the atoms according to the conditions specified by the user in the interface, and outputs the results to coverage files for later examination and editing.
- An interface and application for specifying the files containing the finalized VA redistributions for the EDs and combining them into a new boundary set in INDEA.

This document describes the proposed interfaces and applications involved in the IDM automated redistribution components to be developed.

A diagram illustrating the general structure of the interfaces and application components of the proposed IDM automated redistribution system is shown in Figure 1.

Data Requirements

Database Layers for Atom Generation

The building blocks for the VAs during automated redistribution are a coverage of polygonal atoms generated from specific layers in INDEA. Some of these layers (the "standard" layers) are always used in atom formation, while other layers may be optionally included.

The standard layers are as follows:

- existing VA coverage, for a specified boundary set
- municipal, IR, and Regional District boundaries
- INDEACC boundaries

[2] This is a BCS ASCII file format for representing spatial data in an effective and accessible way. The functionality will be provided in the IDM to read the data from a mapEdit atom file into an IDM layer and save it to a standard IDM coverage file.

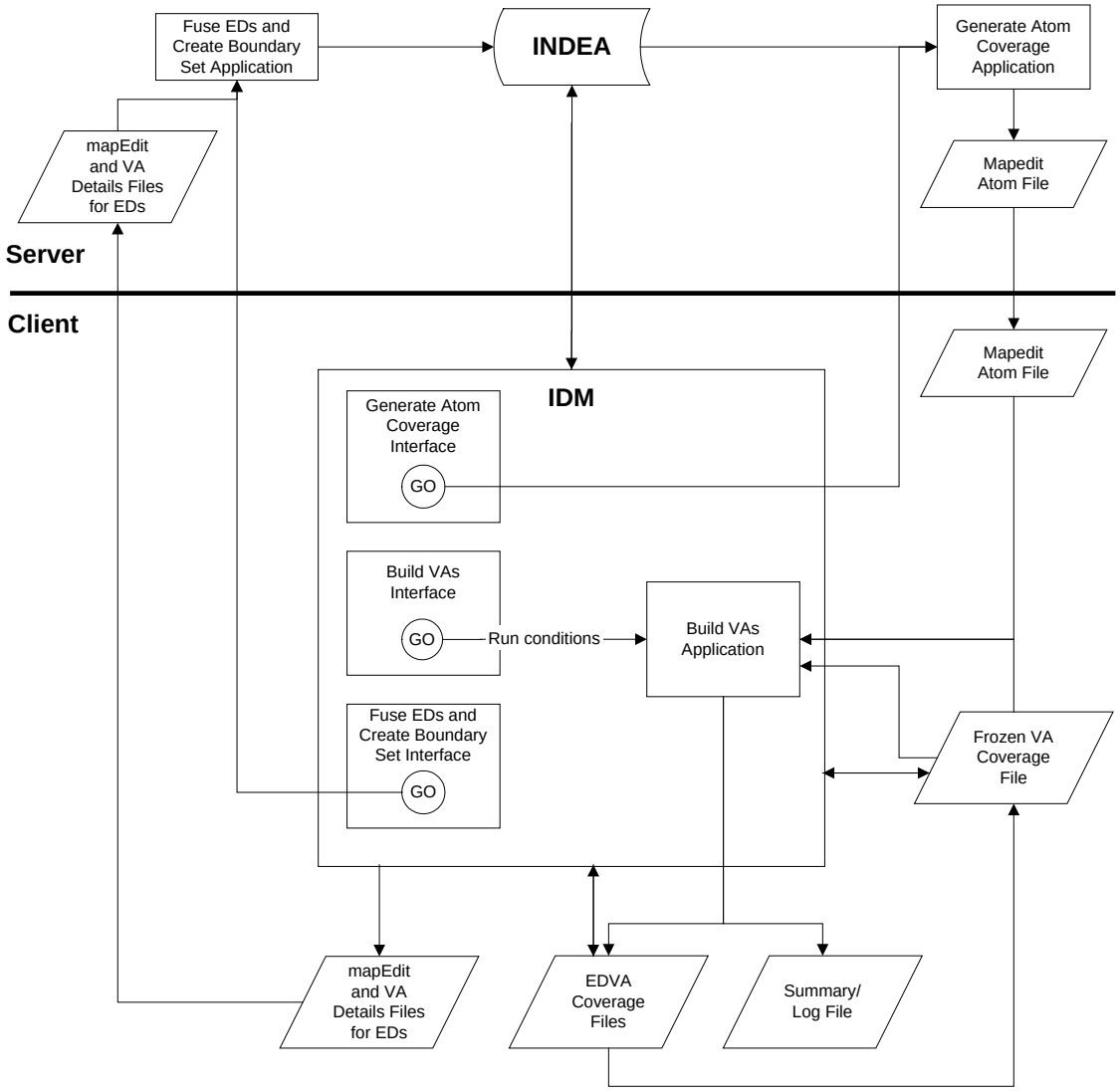

Figure 1. Diagram illustrating the processing and data relationships for the automated redistribution tools of the IDM, including the interfaces and applications for the three main components of the automated redistribution and the organization of server and client processes.

- roads
- non-crossable boundaries (a new layer to contain obligatory VA boundaries)

The optional layers are as follows:

- hydrography (all component line and polygon layers)
- railway lines (including Skytrain)
- hydro lines

In addition, a second new layer, namely, atom-grouping polygons (which is not used to form boundaries but is used to attribute certain atoms) is also used in automated redistribution.

Most of the above layers are familiar INDEA layers which have been rectified (or made consistent) with each other in INDEA. There is one new attribute, however, that will be provided for the features of these layers and should be set appropriately. This attribute indicates whether this feature in an existing layer can be crossed by a VA, i.e., whether this feature must form a VA boundary. Users must ensure that all features in existing layers that are to form VA boundaries have this attribute set appropriately. This may be done using the IDM, by selecting those features in the active layer that are to be made non-crossable and invoking the Make non-crossable item in the Edit menu (to be developed).

The roles of the two new layers are as follows

- ***The Non-crossable boundaries layer*** is used to provide explicit linework that must always form a VA boundary in any redistribution, but which does not correspond to features in existing layers (whose non-crossable attributes could set to indicate that a VA should not cross the feature). Thus, there are two ways of providing information about which INDEA layer feature boundaries must not be crossed by a VA: (1) setting the Non-crossable attribute for the feature in an existing INDEA layer, or (2) including the linework for the feature in the Non-crossable boundaries layer. The user should use this layer to contain that linework that must form a VA boundary but is not present in any other layer (e.g., the centerline of a major lake).
- ***The Atom-grouping polygons layer*** is used to specify a region within which all the atoms should be grouped together into a single VA. The user should ensure that a polygon is provided to surround any region which should not be split into multiple VAs.

Both the above two layers are optional – if no features in these layers are present in the ED being processed, atoms will still be formed and the automated redistribution process can still proceed.

Once the INDEA data in the above layers has been finalized, the process of automated redistribution can begin. As noted above, this process consists of three stages: atom generation, VA building, and incorporation of the results into INDEA. These topics are described in the following sections.

Atom Generation

This process allows the user to specify the data to be extracted from INDEA and used for atom formation, and results in the creation of an atom coverage in the form of an output

mapEdit file in a specified directory on the user's computer. These atoms are then used as input for the next stage of automated VA building.

Interface

The user interface for atom generation is shown in Figure 2. It has the following controls:

Control	Description
ED	This dropbox specifies the ED for which the INDEA layers are to be extracted and the atoms are to be formed.
Boundary set	This dropbox specifies the boundary set for the ED and VAs to be extracted and the atoms to be formed.
Optional layers	This listbox allows the selection of any optional layers, in addition to the standard layers, that should be included in the atom formation.
Include point VAs in voter count	This checkbox allows the voters in point VAs to either be included in the voter count for the atom or not to be included. The default is **not** to include them.
Use atom-grouping polygons	This checkbox allows any atom-grouping polygons to be used to attribute the atoms during the atom generation. The default is to use these polygons and set the atom attributes accordingly.
Group single-address atoms together	This checkbox allows atoms containing roads with the same single-address information to be used to attribute the atoms during the atom generation. The default is to perform the grouping and set the atom attributes accordingly.
Output atom coverage file	This textbox specifies the directory and file name for the resulting mapEdit file of atoms to be placed. A default name will be provided when the ED and boundary set are specified.
Browse	This button allows the file browser to be used to specify the directory and file name for the output mapEdit file.
Generate atom coverage	This button invokes the application that performs the layer extraction, generates the atoms, and transfers the resulting mapEdit atom file to the specified directory on the user's computer.

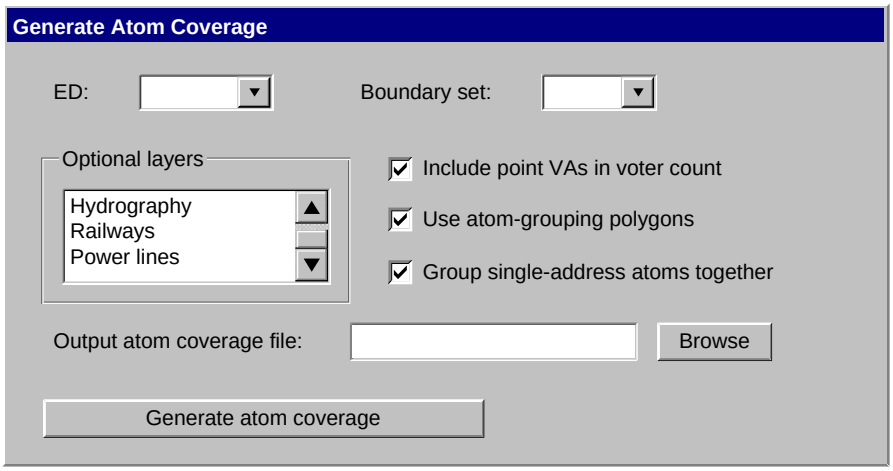

Figure 2. The IDM interface for generating an atom coverage from layers in the INDEA database.

Application

When the conditions have been specified and the Generate atom coverage button is pressed, the IDM first closes the Generate Atom Coverage window and displays a status window with an Abort button. It then sends a message, along with the input conditions, to a program on the INDEA database server. This server program then:

- Extracts the data for the specified layers.
- Combines the linework for the various layers into a coverage of atoms, using an approach similar to that for reconciliation.
- Determines the voter count for each atom in the coverage, taking the point VA voter count option into account.
- Determines the number of residential addresses with zero voters for each atom in the coverage.
- If the atom-grouping polygon option was selected, sets an attribute of the atom to indicate that it should be grouped together with other atoms with the same value for this attribute.
- If the single-address grouping option was chosen, examines the roads within each atom, identifies those atoms that contain roads with the same single-address characteristics, and sets an attribute of the atom to indicate that it should be grouped together with other atoms with the same value for this attribute.
- Writes the final attributed atom coverage to a mapEdit file on the INDEA database server.
- Transfers this mapEdit file to the specified destination directory on the user's computer.
- Sends a completion message to the IDM, which displays a notification message for the user.

VA Building

This process allows the user to specify and perform multiple runs of automated redistribution of the VAs within the ED, during each of which a separate set of VAs is built from the atoms and output to a separate EDVA coverage file.

The application takes two types of input files:

- A mapEdit file containing the atom coverage for the ED (required).
- An EDVA coverage file that specifies the VAs that are to be frozen during the redistribution (optional).

The first file is generated by a run of the atom generation application described above. The second file can easily be generated by extracting the EDVAs for the ED from INDEA (or loading a previously prepared coverage file for the ED) and setting the Frozen-VA attribute for those VAs whose boundaries are to remain fixed during the VA building stage of the automated redistribution runs. Note that the EDVA coverage file can itself be a coverage file resulting from a run of the VA building application. This allows iterative refinement of the result of automated redistribution, by specifying those VAs which are to remain unchanged while allowing the others to be built up again from atoms during successive runs.

The application produces two types of output files:

- A set of EDVA coverage files, each of which contains a separate VA "tiling", i.e., a different grouping of the atoms into VAs.
- A summary/log file containing information about the run conditions, and containing a summary of the results for each VA tiling. This information includes:
    - the number of VAs generated for that tiling;
    - a measure of the overall compactness of the VAs generated for the tiling;
    - a histogram of the voter count distribution of the VAs in the tiling.

The user can then examine the summary/log file, decide which tilings are most likely to be satisfactory, and examine the corresponding coverage files. A VA tiling of interest can then be edited (by loading its coverage file into the IDM) and selected VAs can be frozen if desired. The resulting frozen-VA file can then be used as input to another run of VA building to generate a new set of candidate VA tilings, which can in turn be examined and edited. This iterative process of alternating automated and manual stages ultimately leads to the definition of an acceptable final tiling for the ED.

When the iterative process has resulted in a final satisfactory redistribution being achieved for the ED, the results can be saved in a special format (see below) and used for the next stage of incorporating the new VAs, along with those for other EDs, into a new boundary set in the INDEA database.

Interface

The user interface for atom generation is shown in Figure 3. It has the following control:

Control	Description
Input atom coverage file	This textbox specifies the file name and directory for the input mapEdit file of atoms to be used in the VA building.
Input frozen-VA coverage file (optional)	This textbox specifies the file name and directory for the input EDVA coverage file that specifies those VAs whose boundaries are not to be altered by the automated VA building.
Output file base name	This textbox specifies the file name and directory for the summary/log file generated by the run. It also forms the base name for the output EDVA coverage file(s) generated by the VA building application. If multiple files are generated, each will have the same base name, which is appended with a unique tag to indicate the run conditions used for its generation.
Browse	This button allows the file browser to be used to specify the file name and directory for the respective file.
Target voter count (Low, High)	These text boxes allow the user to specify the lower and upper targets for the voter count for the VAs. Addition of atoms to a growing VA stops when the voter count reaches the lower limit. Also, an atom cannot be added to a growing VA if the addition of that atom would cause the voter count for the resulting VA to exceed the upper limit.
Pseudo-voter count	This text box allows the user to specify the number of "pseudo-voters", i.e., the number of potential voters that are assumed to be present in each residential address which currently has a zero voter count.
Starting atom selection	These radio buttons allow the user to specify the scheme to be used for selecting the starting atom for each new VA as it is built. The outside-in strategy chooses the most distant unassigned atom from the attractor atom, while the linear-march strategy chooses the next unassigned atom along a specified direction. The interface allows for the specification of multiple runs with either strategy. For the outside-in strategy, the user specifies the number of attractor atoms (the choice of which atoms will be the actual attractor atoms is made automatically by the algorithm). For the linear-march strategy, the user specifies an initial and final direction (degrees, clockwise from North) and an increment.
Build VAs	This button invokes the application that performs the VA building and generates the summary/log file and the coverage file for each separate VA tiling.

The run conditions specified in the interface can be saved in a configuration file using the IDM File menu item.

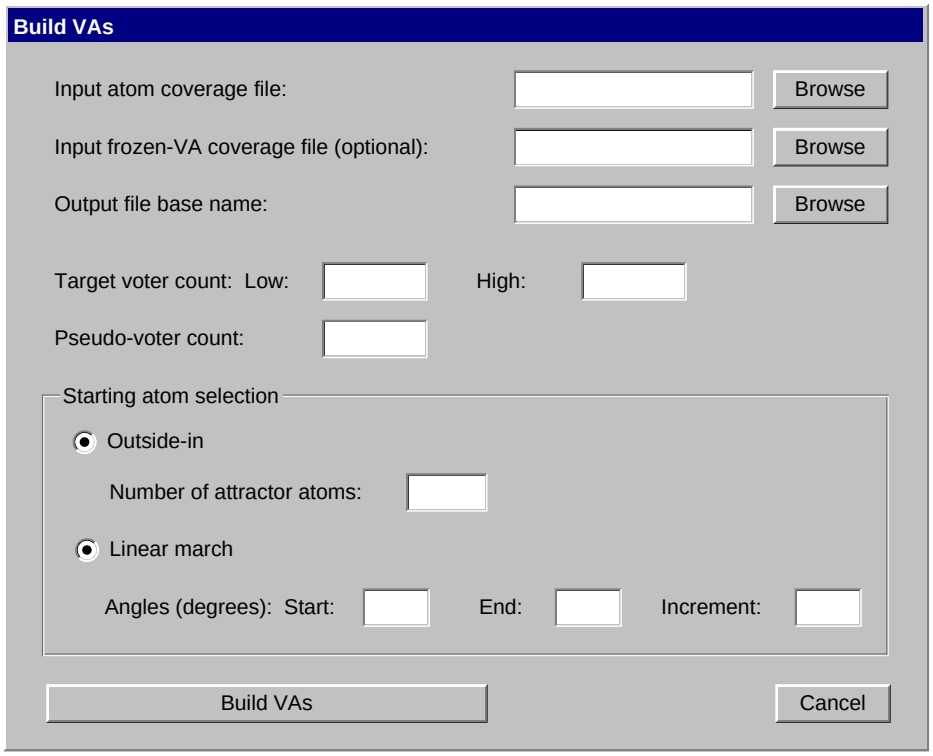

Figure 3. The IDM interface to control the automated building of VAs from the atom coverage.

Application

When the conditions have been specified and the Build VAs button is pressed, the IDM first closes the Build VAs window and displays a status window with an Abort button. It then invokes an internal IDM process, which:

- Inputs the atom coverage file and the frozen-VA file (if specified).
- Performs a separate run for each set of input conditions specified by the user.
- Outputs the EDVA coverage file containing the VA tiling for each run.
- Outputs a summary/log file containing input conditions and statistics for the entire set of runs.
- When finished, displays a completion message for the user. (Alternatively, a progress bar could be displayed while the application is running.)

Fusing EDs into a New Boundary Set

This process allows the user to select one or more EDVA coverage files, each of which contains the final results of the redistribution for an ED, and to fuse them together to generate a new boundary set in the INDEA database.

When the redistribution for each ED is completed, the user should save the results into a "ready" directory that will contain the files for the completed EDs (and only those files) in a special format. These files must be saved using the "Save As Fuse Files" option of the IDM Save menu item, which creates a mapEdit file and a VA details file for the ED. These files are in the format required by the Fuse EDs and Create Boundary Set application.

When the redistribution processing has been completed for all the selected EDs, the user should ensure that the files for all the EDs that have been processed and are to be fused are in the "ready" directory. The IDM fusing/incorporation process can then be used to fuse the data for the EDs in these files and create a new INDEA boundary set. During this process, the data for any EDs not represented in the "ready" directory (i.e., any EDs that were not redistributed) will be obtained from the current EDVA boundary set, so that the resulting boundary set will be a full coverage for all BC.

Interface

The user interface for fusing EDs into a new boundary set is shown in Figure 4. It has the following controls:

Control	Description
Directory for input files	This textbox specifies the directory for the files containing the redistributed EDs that are to be fused and used to create a new boundary set.
Browse	This button allows the file browser to be used to specify the directory containing the files for the redistributed EDs.
Boundary set number	This textbox allows the user to specify the number for the new boundary set to be created.
Boundary set name	This textbox allows the user to specify the name for the new boundary set to be created.
Fuse EDs and create boundary set	This button invokes the application that performs the file transfer, fuses the EDs and creates the new boundary set.

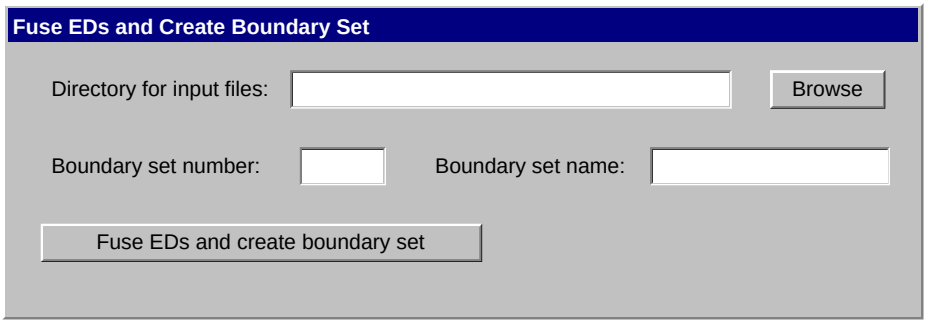

Figure 4. The IDM interface to fuse redistributed EDs and combine them into a new boundary set.

Application

When the conditions have been specified and the Fuse EDs and create boundary set button is pressed, the IDM first closes the Fuse EDs window and displays a status window with an Abort button. It then sends a message, along with these input conditions, to a program on the INDEA database server. This server program then:

- Transfers the selected EDVA mapEdit and VA details files from the "ready" directory to the server.
- Invokes a program that reads in the mapEdit and VA details files and fuses their contents into the form of a full and internally consistent EDVA coverage for all BC, using data from the current boundary set to populate any EDs that were not present in the "ready" directory.
- Creates a new boundary set in INDEA and populates it with the new EDVA coverage.
- Sends a completion message to the IDM, which displays a notification message for the user.